



Manualens gennemgang af de enkelte kommandoer er forbilledlig. Dog savnes et brugbart index.

COMAL for voksne

Comal er blevet voksen.

Med lanceringen af IBM Comal 80 har Comal fået et vigtigt blåt stempel. IBM Comal 80 er hurtig. Den er brugervenlig. Alt, hvad erhvervsprogrammøren kan ønske sig. Spørgsmålet er, om det er nok. Det har "Alt om Data's" John Christoffersen nogle meninger om.

■ Comal er et skolesprog. Comal er for langsomt og simpelt til seriøse programmer. Det er nogle af de synspunkter man hører, når man kommer for skade at nævne Comal i selskab med professionelle erhvervsprogrammører. Og sandt at sige har det hidtil knebet med gode forsvar. Det nytter jo ikke stort at fremhæve f.eks. Commodore Comal's fremragende faciliteter, når programmøren er sat til at lave en professionel brancheløsning til en PC'er.

Vel har der været gjort forsøg på at implementere Comal til 16-bit systemer. Regnecentralens RC-Comal til CP/M 86 (og den videreudviklede version til Gyldendals Scandis) samt Mytech Comal til MSDOS systemer.

Men ingen af de nævnte har haft nogen særlig gennemslagskraft – undtagen i skolesammenhæng, hvor behovet for pædagogiske fejlmeldinger og enkelthed langt overstiger kravene til datakraft og faciliteter.

Men måske kan fremkomsten af IBM Comal 80 vende billedet totalt. For her er der virkelig tale om datakraft og faciliteter, der burde få en og anden kronisk BASIC- eller Pascal fanatiker til at løfte lidt på de tunge øjenlåg.

IBM er ikke ligefrem kendt for at lave pædagogisk legetøj. Og når de har otret systemfolk og ressourcer på udviklingen af IBM Comal 80, er det ikke bare for at tilfredstille det relativt lille segment af det danske skolemarked, som har råd til MS DOS løsninger. Spørgsmålet er så bare, om de konservative danske systemprogram mører lader sig overbevise. Det vil kun fremtiden – og salgstallene – vise.

Danske Unicomal står bag.

I stedet for at sætte egne programmører til at udvikle Comal 80 fra bunden, har IBM klogeligt valgt at sætte Unicomal folkene på sagen. Deres arbejde med 6502-processor baseret Comal som i den seneste Commodore version er ikke gået upåagtet hen i de flotte kontorer i Lundtofte. Men IBM har haft naturligvis haft en hel del krav til programmet, som har sat sig spor. F.eks. er PC DOS blevet tilgængeligt direkte fra Comal 80, hvilket vil sige at direktiver til printer, som i standard Comal 80 hedder "lp:" i IBM versionen hedder "pm:". Men det er også det eneste sted, hvor overspilning af

program mellem f.eks. en IBM PC og en Commodore 64 vil give problemer.

Testudgaven havde versionsnummeret 2.0. Og det er noget slemt sludder. Unicomal har kun lavet den samme version, så den burde hedde version 1.00 – bare sådan. Ideen er selvfølgelig at henlede opmærksomheden på Commodore Comal 80 version 2.00, der med rette regnes for det fremmeste Comal-80 produkt, når det gælder features og faciliteter. Og naturligvis er IBM fuldt kompatibel med kapslen – men alligevel...

IBM har også haft krav til hastighed. Og fuld kompatibilitet med diverse IBM-grafikkort og vedtagne systemadresser. Ingenlunde nogen let opgave, PC-DOS's træghed i skærmopbygningen taget in mente. Men opgaven tør siges at være løst til UG.

På det anvendte testudstyr – en Olivetti PC – klarede Comal 80 den vante 8 punkts benchmark-test med et gennemsnit på 4.89 sekunder. Det bringer den på højde med en 32 bit supermicro som Sage II – med BASIC under p-code systemet. En hastighedstest på en IBM PC viser et gennemsnit på lige under 10 sekunder. Stadig næ-



Comal bruges ikke kun som undervisningssprog i skolerne. Også professionelle programmører, er begyndt at bruge Comal. Her på virksomheden Dansk System Elektronik i Horsens.

sten dobbelt så hurtigt som BASICA. Og har man en 8087 matematikprocessor indbygget – Comal 80 leveres med en fuldt implementeret 8087-version på disketten – går det helt amok: En Olivetti med 8087'er klarer 8 punkts testen med lidt over 1 sekund i snit – en tilsvarende IBM med lidt over 2 sekunder. Pow!

Begrænsninger og muligheder med IBM Comal 80.

Det er nærliggende at sammenligne IBM Comal med Commodore 2.00 versionen. Men en given PC har nogle begrænsninger i forhold til Commodore'n – primært hvad angår grafik og lyd. Derfor er der ikke nogen Sprite- og Soundpakker i IBM Comal. Ligesom der heller ikke er pakker til læsning og styring af joysticks, paddles med videre.

Til gengæld har IBM nogle hukommelsesmuligheder, som IBM Comal udnytter til fulde. Som standard kan IBM Comal adressere 64K program og 64K data. Men så er der desuden muligheder for at bruge externe procedurer – altså programafsnit lag-

ret på disketten eller i den resterende del af memory.

Skriv dit eget GEM i Comal.

Faktisk kan man bruge ekstra hukommelse til lidt af hvert. Man kan have et helt selvstændigt Comal program indlæst, som kan køres fra det oprindelige Comal via kommandoen PASS. Men smartere endnu: Man kan nemlig load og køre alle mulige programmer direkte fra IBM Comal 80 – uden at forlade programmet!

Det gør det muligt at lave et menustyret interaktivt program i IBM Comal og køre dBASE II, WordStar, BASIC og Comal-programmer med mere. Forestil dig et hjemmelavet lagerstyringsprogram, hvor du til udskrift og editring af dokument kalder WordStar – sådan. Det er en virkelig stærk feature – faktisk kræver det kun den ene kommando, PASS "program-navn".

Formattering af Input-sætninger.

En vigtig detalje i stort set ethvert erhvervsprogram er check af fejl – typisk de fejl, som brugeren kan generere ved Input. I IBM Comal 80 kan Input sætninger beskyt-

tes – direkte ved angivelse af parametre til Input "spørgsmål" sætninger. Dog kan en sådan kommando blive lovlig lang og uoverskuelig, for eksempel Input "spørgsmål" AT x,y,længde,prompt,variabler,mærke – og det er stadigvæk kun muligt at forhindre uønskede strenginput til en strengvariabel ved at checke for del bagefter – men det er da en start. Naturligvis kunne man ønske, at Input sætninger kunne formateres ligeså nemt som Print Using "**** *" – men det må komme i version 2 – undskyld – 3.00.

Interaktiv fejlfinding hjælper programmøren.

En af de rareste ting ved programmeringsarbejde med Comal er de nemme og beskrivende fejlmeldinger, som kører interaktivt med programmøren under arbejdet. Syntakscheckereren kører på højeste gear konstant, og generelt er det stort set umuligt at lave en syntaksfejl i en sætning. Strukturfejl findes med Scan, som sørger for, at antallet af Proc-Endproc'er, For-Enfor osv. passer sammen. Uønsket variabel genbrug findes også – det er umuligt at gendefinere en allerede oprettet varia-

bel. Selv om de tilladte 238 karakterer pr. variabelnavn giver astronomiske muligheder for at undgå fejltagelser. Det sidste gav i øvrigt anledning til en sjov fejl. I Comal må man ikke have flere variabeltyper med samme navn. F.eks. er Dato = val(Dato\$) forbudt. En ellers meget nyttig ting, eftersom indholdet i den numeriske Dato og strengvariablen Dato\$ er det samme. Her er fejlmeldingen imidlertid ikke v\$: Variabel allerede oprettet. Comal siger v\$: Forkert type, hvilket er noget logisk nonsens. Men det skal rimeligvis tilføjes, at det hører til i petittesse-afdelingen.

Programstyret fejlhåndtering

I selve programmet er det muligt at fange fejlene – og give forskellige muligheder for at reagere på dem under programstyring. Eksempelvis forsøg på at læse/skrive til ikke tilsluttede enheder, hente flere data, end der er oprettet – det kan alt sammen gøres uden at Comal bryder ud på systemet. Trap-rutinerne indeholder mulighed for at bruge Comal's fejlmeldinger eller definere sine egne. Fil og strømnumre ved f.eks. diskettefejl kan gemmes i en systemvariabel og der er endog

COMAL for voksne

mulighed for at bruge Retry funktionen og dermed give brugeren en chance for at rette fejlen – tænde for printeren, indsætte en diskette i drev b: osv – og bagefter fortsætte programudviklingen som om intet var hændt.

Comal opfylder IEEE standarder

IEEE står for Institute of Electric and Electronics Engineers – en sammenslutning, som har vedtaget en række internationale standarder, blandt andet også for computere.

En af disse normer gælder nøjagtighed for udregninger. Dem opfylder IBM Comal 80 til fulde – alle udregninger udføres med 16 cifres nøjagtighed. Endnu en grund til at betragte benchmarkresultaterne med en vis ærbødighed.

IBM Comal 80 kan håndtere tal fra minus til plus $1,7 \cdot 10^{308}$ – det er et ganske stort tal. Det gør det interessant også at anvende IBM Comal 80 i stærkt specialiserede sammenhænge, der kræver disse uhyrlige talstørrelser.

I heltalsvariabler kan IBM Comal 80 bruge tal på op til 10 cifre. Igen væsentligt mere end almindeligt.

Grafikken kan styres fra Logo. Til styring af grafik er der det fulde Logo-kommandosæt til rådighed. Og det er bestemt ikke noget at rynke på næsen ad, når man ser, med hvilken hastighed det udføres på en PC.

Grafikken er indlagt som en "pakke". Det vil sige, at kommandosættet først er til rådighed, når kommandoen Use Graphics er givet i programmet. Angiveligt for, at Logo-kommandoerne og de tillad-

te forkortelser ikke skal generere eventuelle variabler i et program genereret på en Comal uden grafik.

Det er også muligt at definere vinduer og nye koordinatsæt skærmen (så man eksempelvis kan udskrive grafer af sinusresultater uden at multiplicere værdierne af hensyn til udlæsningen) med så simple kommandoer som Viewport og Window.

Ændre defaults undervejs.

Pakken "system" indeholder flere muligheder for at ændre i IBM Comal 80's standardværdier. Forgrunds/baggrundsfarver ved fejlmeldinger, indskrivning af tekst med videre. Desuden fås en række specialkommandoer, hvorefter man kan ændre f.eks. strengvariabler til udelukkende at indeholde store (Upper\$) eller små (Lower\$) tegn.

Det er også herigennem, man kan definere funktionstasterne (1 til 40 muligheder) samt læse værdier fra diverse porte, afhængigt af op de er tilsluttede eller ej.

Comal kan udbygges med biblioteker.

Comal er opbygget i moduler. Mellem bruger og program findes en såkaldt navnetabel, der dels indeholder oplysninger om, hvilke kommandoer og procedurer, der er til rådighed – dels, hvor i Memory, de kan findes.

Det gør det forholdsvis nemt tilgængeligt at udvide sproget med "pakker" og nye kommandoer – uanset om de er skrevet i maskinkode eller Comal selv.

En pakke, skrevet i det rette

format, kan Use's, hvorefter dens kommandosæt indlemmes i navnetabellen. Dermed er der mulighed for at opbygge hele biblioteker af kommandoer – eller helt selvstændige programmer – som så kan integreres i Comal og bruges når – og kun når – det ønskes i en given sammenhæng.

Et eksempel, jeg kender til, er Dansk System Elektronik i Horsens, der bruger IBM Comal 80 med diverse ekstrakommandoer til styringsformål i automatiserede produktionsskemaer. Med en kommando kan de læse resultaterne fra en given føler på en værkstedsmaskine og med en kommando give nye direktiver til maskinen.

Det er systemprogrammering, så det er til at forstå. Og det sparer virksomhedens programmerer bunker af tid og ærgrelser at have et sådant specialtilpasset sprog til rådighed.

God og overskuelig manual.

Manualen til IBM Comal 80 skal betragtes som et opslagsværk og ikke en lærebog i Comal 80. Og helt fint er der angivet en særdeles omfattende litteraturliste over egnede lærebøger.

Alle kommandoer er gennemgået med et eller flere programsekvenser til belysning af de særlige finesser ved betydningen – og de er ydermere næsten allesammen at finde på Comal 80 – disketten som demo'er – lige til at hente.

Dog er der en ting, som ærgrer mig. Som opslagsværk er det udmærket, at kommandoer er alfabetisk listet. Men

IBM Comal gør god brug af 80 tegns skærmformat. Det er tilladt at skrive sætninger på helt op til 240 karakterer.

```
(C) Copyright IBM Corp 1984
(C) Copyright UniComal 1984

load "fac"
list
0010 FUNC fakultet(x) CLOSED
0020 TRAP
0030 RETURN fac(x)
0040 HANDLER
0050 RETURN 1.0E+300
0060 ENDTAP
0070 //
0080 FUNC fac(n)
0090 IF n=0 THEN
0100 RETURN 1
0110 ELSE
0120 RETURN n*fac(n-1)
0130 ENDDIF
0140 ENDFUNC fac
0150 ENDFUNC fakultet

LIST 2 RUN+ 3 AUTO+ 4 RETURN+ 5 DIR+ 6 EDIT 7 CON+ 8 LOAD 9 FIND 10 SEL COM
```

det er til gengæld en forsømmelse, at index'et stort set kun indeholder de samme kommandoer – en alfabetiseret udgave af indholdstegnel sen. Eksempelvis kan man ikke finde ordet "Printer" i index'et. Man skal vide, hvor man finder de printerrelaterede kommandoer.

Til gengæld er der mange nyttige oplysninger at hente i de mange appendiks'er. Alt hvad programmøren har brug for at vide om programmets virkemåde, fejlmeldinger, linkninger af maskinsprog, oprettelse af nye pakker og så videre.

Konklusion

Comal er kommet et lang stykke vej siden de glade pionerdage med Børge Christensen og drengene i Tønder. IBM Comal 80 er et fremragende, interaktivt programmeringsværktøj, der gør udvikling af programmer til de fleste formål legende let. Samtidig er IBM Comal 80 overbevisende hurtigt. Så hurtigt, at det må betyde en kraftig revurdering af compilerede sprogs fortræffeligheder i mange af de applikationer, professionelle programmerer sidder og sveder over til dagligt.

Især Comal 80 – 8087 processor versionen, der leveres standard med systemet, rummer store muligheder for udvikling af barske talknuser-programmer. I et miljø, der er både hjælpsomt og venligt. Til en pris af under 3000,- kr (plus moms) er IBM Comal 80 endda nærmest provokerende billig, set i forhold til andre professionelle IBM-programmer. □

John Christoffersen